The University of Chicago
Center for Integrating Statistical and Environmental Science
www.stat.uchicago.edu/~cises

Chicago, Illinois USA

TECHNICAL REPORT NO. 31

STATISTICAL CONDITIONAL SIMULATION OF A MULTIRESOLUTION NUMERICAL AIR QUALITY MODEL

Xiaofeng Shao, Michael L. Stein

December 2005
Revised April 2006

Although the research described in this article has been funded wholly or in part by the United States Environmental Protection Agency through STAR Cooperative Agreement #R-82940201 to The University of Chicago, it has not been subjected to the Agency's required peer and policy review and

therefore does not necessarily reflect the views of the Agency, and no official endorsement should be inferred.

Statistical conditional simulation of a multiresolution numerical air quality model

Xiaofeng Shao and Michael Stein[1]

Abstract: This paper addresses sub-grid variability, an issue that naturally arises in multiresolution numerical air quality models. Unlike previous approaches, which fit a parametric distribution over a spatial block and perform the fit from block to block independently over space and time, our approach to dealing with the sub-grid variability is to describe the space-time conditional distribution of high resolution output given its low resolution counterpart. A novel conditional simulation approach is proposed to produce an ensemble of high resolution runs based on the runs we have, and various criteria are used to assess whether our simulated high resolution runs capture the overall space-time variability of the original high resolution runs. The main idea of our algorithm is to apply a nonlinear filter to the high resolution runs based on the low resolution runs, then perform a time domain block bootstrap for the residuals simultaneously over space. The algorithm proposed in this paper can be readily used by practitioners to generate random high resolution runs as a useful surrogate to the real high resolution runs when one has low resolution runs for a long period and only a few days' high resolution runs.

1 Introduction

Numerical air quality models play an important role in the study of air pollution and regulation of environmental protection. Currently, the Models-3/Community Multiscale Air Quality model (hereafter, CMAQ) [cf. Byun and Ching (1999)] is one of the main air quality simulation models the US Environmental Protection Agency (EPA) uses. It serves as a basis for implementing the national ambient air quality standards and a tool for performing risk-based assessments. It is also the model used for air quality forecasting by EPA and NOAA (National Oceanic and Atmospheric Administration). The simulation of air pollution by CMAQ often involves runs at multiple resolutions. The aggregation of a high resolution run is not simply the corresponding low resolution run. Some issues of comparison of CMAQ runs at different resolutions have been addressed in Shao, Stein and Ching (2005). The main goal of this paper is to address another important issue related to runs at multiple resolutions, the within-grid variability (hereafter, sub-grid variability) [Ching et al. 2004, 2005]. Sub-grid variability is an

[1]Xiaofeng Shao is Doctoral Student and Michael Stein is Professor in the Department of Statistics, University of Chicago. 5734 S. University Avenue, Chicago, IL 60637 (E-mail: shao@galton.uchicago.edu, stein@galton.uchicago.edu). Although the research described in this article has been funded wholly or in part by the United States Environmental Protection Agency through STAR Cooperative Agreement #R-82940201-0 to the University of Chicago, it has not been subjected to the Agency's required peer and policy review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

important factor in various areas such as surface hydrology [Chan et al. (1997)], precipitation modeling [Harris and Foufoula-Georgiou (2001)], and climate downscaling [Wood et al. (2004)]. The main focus of this paper is on the sub-grid variability of the multiresolution CMAQ output.

To assess accurately the human risk due to the exposure to hazardous air pollutants at neighborhood scales, fine-scale information on ambient air toxic concentrations is needed. Typically, this kind of fine-scale information cannot be provided by sparsely collected monitoring data, so high resolution output from a numerical air quality model like CMAQ is a potential alternative. Currently, CMAQ simulations at fine resolution (1-2km grid spacing) can be implemented on a regional scale thanks to improvements in computing power and the refinement of nested grid techniques. These available high resolution runs provide an opportunity to study the sub-grid variability of lower resolution runs. Figure 1 shows the runs at 2km and 8km resolution for carbon monoxide (CO) at 1am on Aug 1st, 1999 (see Section 2 for a detailed introduction to the data). By construction, the variability at 8km resolution is ignored in the 8km$\times$8km grid box of 8km resolution output. Since there is always a finer resolution for a given resolution (say, 2km resolution), the sub-grid variability is defined in a relative sense. In this paper, our focus is on the variability within the 8km$\times$8km grid box that is ignored in the 8km resolution runs. To study the sub-grid variability, there have been some efforts in modeling the marginal distribution of a high resolution run within a low resolution grid [Herwehe et al. (2004), Ching et al. (2005)]. For example, Herwehe et al. (2004) used the Weibull distribution to fit the values from a high resolution run for each low resolution grid independently from grid cell to grid cell over time. These fitted marginal density functions are produced to represent the sub-grid variability, which are subsequently used as input data for hazardous air pollutant human exposure models. This procedure ignores the spatial and temporal dependence in the processes, which, we contend, is essential to understanding the sub-grid variability and the difference between the runs at two resolutions. In this paper, we argue that a more sensible way to approach the sub-grid variability issue is to ask the following question: What is the conditional distribution of a high resolution run given the run at low resolution of the same time period? For random quantities X and Y, denote by $X|Y$ the conditional distribution of X given Y. We are interested in $\{Z_H(t), 1 \leq t \leq T\}|\{Z_L(t), 1 \leq t \leq T\}$, where $Z_H(t)$ and $Z_L(t)$ stand for the high resolution and low resolution runs at time t, T indicating the total number of hours involved.

To describe the conditional distribution, one could model it parametrically. Considering the fact that any detailed parametric model we might develop would be specific to the area and the time period under consideration, we instead propose an algorithm to conditionally simulate the high resolution runs based on the runs we have in a way that could be relatively easily applied to other times, places and pollutants. The simulated high resolution runs have approximately the same conditional distribution as the original high resolution runs given the corresponding low resolution runs. Conditional simulation under a spatial or space-time framework is not a new topic. See Chilès and Delfiner (1999) for a detailed introduction to

conditional simulation in a geostatistical context. Recently, in a space-time setting, Nunes and Soares (2005) introduced a simulation methodology to characterize air pollution dispersion. Our simulation algorithm is proposed under a space-time multiresolution framework and it seems that there is no previous work that addresses this problem. The most closely related work in the literature is on precipitation downscaling (or disaggregation) [cf. Perica and Foufoula-Georgiou (1996), Venugopal et al. (1999), Harris and Foufoula-Georgiou (2001)]. However, an essential difference is that their goal is to produce the right spatial marginal variability but ours is to reproduce the overall space-time variability. Moreover, the specific approaches are different.

The paper is structured as follows. Section 2 introduces our data and states the problems explicitly. Section 3 describes our algorithms for in-sample prediction and out-of-sample prediction. Section 4 considers the evaluation of our simulations by various criteria. Section 5 describes an alternative simulation method that produces better out-of-sample prediction. Section 6 provides some discussion and conclusions are drawn in Section 7. Some formulas are left to the appendix.

2 Data and Problem Statement

2.1 The Data

We have hourly CMAQ runs at three different resolutions over the Atlanta area. These runs were provided by the US EPA. The finest resolution has grid spacing 2km and the resolution decreases by a factor of 4, i.e. the two low resolution runs have grid spacing 8km and 32km respectively. For each resolution, we have hourly output from Aug 1st to Aug 24th, 1999. In our study, we take a subset of the outputs from three resolutions over their overlapped domain, which corresponds to 96×112, 24×28 and 6×7 cells, respectively. Here the first number corresponds to the number of grid cells in the south-north direction and the second is the number of cells in the east-west direction. Denote by $Z_2(t)$, $Z_8(t)$ and $Z_{32}(t)$ the runs at time t at 2km, 8km and 32km resolutions. Here, $t = 1, \cdots, T = 576$, since we have hourly output for 24 days. In this paper, we focus on the species CO. The performance of our algorithm for other species will be discussed in Section 7.

2.2 Problem Description

Currently, CMAQ runs at multiple resolutions are performed using "One Way Nesting" [Byun and Ching (1999)]. Specifically, to run CMAQ at 2km resolution, one needs to first get the run at 8km resolution, an interpolation of which provides the initial and boundary conditions for the run at 2km resolution. In a sense, we can think of the CMAQ run at 8km resolution as "input" and the run at 2km resolution as "output". Of course, the relationship between "input" and "output" is highly nonlinear and depends on factors such as meteorology, chemistry and

emissions. CMAQ output is deterministic given its inputs. A possible way to address the sub-grid variability is to perturb the low resolution run ("input"), run the high resolution model with initial and boundary conditions based on the interpolation of perturbed low resolution runs, then use the resulting high resolution runs as a probabilistic ensemble. This approach is relevant to our question and similar ideas have been applied in the literature of ensemble weather forecasting [see Hamill and Colucci (1997) and the references therein]. However, it is difficult to carry out since running CMAQ at high resolution is very computationally demanding and it is not clear how to perturb the low resolution run appropriately. Alternatively, one can perturb some aspect of the high resolution runs, but should one perturb meteorology, emissions, or the physics or chemistry within CMAQ?

Another approach to sub-grid variability is to model the conditional distribution statistically based on the runs we have. However, since any parametric statistical model we might develop here would be specific to the dataset under the study, we instead propose an algorithm to conditionally simulate the high resolution runs based on the runs we have. The algorithm we propose can be easily applied to other times, places and pollutants. In particular, we divide our runs into two sets, a training set, which contains the runs from Aug 1st-Aug 12th, 1999 and a testing set, which covers the runs from Aug 13th-Aug 24th, 1999. In this paper, we use $Z([t_1, t_2])$ to denote $\{Z(t), t_1 \leq t \leq t_2\}$, where for each t, $Z(t)$ could be a spatial process or other vector-valued process. Two basic problems are addressed in this paper, namely, in-sample (or in-sample prediction) and prediction (out-of-sample prediction). In the in-sample procedure, we simulate a probabilistic ensemble of high resolution runs of the training set $Z_H^{(j)}([1, T_1]), (T_1 = 288)$ based on the runs at two resolutions from the training set, i.e. $Z_L([1, T_1])$ and $Z_H([1, T_1])$. Here the superscript j indicates the jth simulation and its value ranges from 1 to 100 since 100 simulations were conducted. The in-sample simulation is a way to check the internal validity of our algorithm. In the prediction problem, based on the low and high resolution runs from the training set $Z_L([1, T_1]), Z_H([1, T_1])$ as well as the low resolution run from the testing set $Z_L([T_1 + 1, T])$, we produce an ensemble of $Z_H^{(j)}([T_1 + 1, T])$ to compare with the true high resolution run in the testing set, i.e. $Z_H([T_1 + 1, T])$. The main concern here is whether our simulation can preserve the space-time variability in the original high resolution runs. See Section 4 for more details about the verification and Section 6 about implications for producing spatial maxima and 4-hour spatial maxima. Note that our goal is not to predict a point value in the high resolution output at a given time and a spatial location. It is the reproduction of the overall space-time variability that we care about. This is also the main goal of Nunes and Soares (2005) in a similar conditional simulation problem.

3 Conditional Simulation Algorithm

In this section, we describe our algorithm for conditional simulation. We first introduce the steps involved in the in-sample.

3.1 In-sample

- Step 1. Nonlinear filtering on the training set. For $1 \le t \le T_1$,

$$\log\{Z_H(s;t)\} = a_t + b_t \times \mathrm{BI}[\log\{Z_L(t)\}](s) + r(s;t), s \in S,$$

where $S = \{1, \cdots, 96\} \times \{1, \cdots, 112\}$ is the index set in space. Here $\log\{Z_L(t)\}$ is the pixelwise logarithm of the low resolution output at time t and $\mathrm{BI}[\cdot]$ signifies the bilinear spatial interpolation from 24×28 to 96×112 resolution. For the details of bilinear interpolation and its advantages over other interpolation methods in dealing with CMAQ output, we refer the reader to Shao, Stein and Ching [2005]. Denote by $\hat{a}_t$ and $\hat{b}_t$ the least squares estimates based on the training set and $r(s;t)$ the residual for each (s,t). Taking logarithms of CO concentrations makes both $Z_L(t)$ and $Z_H(t)$ close to normally distributed. Figure 2 shows the average coherence between $\log\{Z_H\}([1,T_1])$ and $\mathrm{BI}[\log\{Z_L\}]([1,T_1])$ (See Appendix for the definition of average coherence between two space-time random fields). We can see a strong coherence at very low frequency ordinates and at $\lambda_k = 2\pi k/T_1, k = 12, 24, \cdots$. This periodic coherence is due to the similar diurnal cycles in the high resolution and low resolution output. It can be seen that the average coherence between the residuals and the bilinear interpolation of low resolution output (at log scale) is substantially lower at low frequency ordinates. A noticeable feature is that at $\lambda_k, k = 12, 24, \cdots$, the coherence is even smaller than at other ordinates, which suggests that the simple nonlinear filtering works to some degree in terms of taking out the diurnal synchrony between the high resolution output and its low resolution version. However, at frequencies close to π, we see that the coherence is strengthened a little bit. Further analysis shows that this phenomenon can be removed if we perform the nonlinear filtering in the frequency domain; see Section 5 for the details. Although we do not expect $\mathrm{BI}[\log\{Z_L\}]([1,T_1])$ to be perfectly independent of the residuals $r([1,T_1]) = \{r(s;t), s \in S, 1 \le t \le T_1\}$, we do see that the original dependence between $\log\{Z_H\}([1,T_1])$ and $\mathrm{BI}[\log\{Z_L\}]([1,T_1])$ has been reduced after the time domain nonlinear filtering.

- Step 2. Simultaneous space-time block bootstrap for the residuals $r([1,T_1])$. The nonparametric block bootstrap method introduced by Künsch [1989] has been widely used in statistics and other fields. For example, it has been applied by Vogel and Shallcross [1996] to the estimation of storage capacity of a surface water reservoir. Suppose we have a stationary time series $X([1,T])$ and wish to generate a random bootstrap sample to mimic the dependence of the original observations. We first specify a block size B, then form $T - B + 1$ blocks where each block is a collection of consecutive observations; for example, the first block is $X([1,B])$, the second block is $X([2,B+1])$ and so on. In the resampling stage, we sample T/B blocks independently with equal probability to sample one of $T - B + 1$ blocks. Finally we concatenate the sampled T/B blocks and form the bootstrap sample $X^*([1,T])$. For notational convenience, we assume the block

size B is divisible by the sample size T in this paper. Note that the dependencies of $X([1,T])$ at small lags are well preserved in the bootstrap sample if B is relatively large. For example, if the original time series is approximately independent beyond five lags, then a block bootstrap sample with appropriate block size (say, 15) can mimic the time dependence of the original time series very well. Also, block bootstrap sampling is very easy to implement. Here we apply this idea to sample from $r([1,T_1])$ simultaneously over space. Empirically, the diurnal cycle of $r(s;t)$ for each pixel s is quite weak. The spatial average of the temporal correlation of $r([1,T_1])$ at lags 1,2,3,4 and 24 are 0.881, 0.712, 0.560, 0.434 and 0.226 respectively. So we let $B = 16$ and the correlation at short lags is well preserved in the bootstrap sample. Since $B = 16$, this algorithm does not capture the correlation at lag 24, which does not seem to affect the overall performance of our algorithm; see Section 4 for more discussion.

In the implementation of block bootstrap, we first get the bootstrapped indexes and then obtain the bootstrap sample by matching the indexes. Denote by $t^*([1,T_1])$ the bootstrapped indexes from the set $\{1,2,\cdots,T_1\}$ with block size B. For example, for a set $\{1,2,\cdots,8\}$ with block size 2, one possible realization of bootstrapped indexes are $\{2,3,1,2,5,6,4,5\}$, where we note that every pair forms a block of size 2. Then the bootstrap sample is $\{X_2,X_3,X_1,X_2,X_5,X_6,X_4,X_5\}$. Denote the bootstrap samples of $r(s;t)$ by $r^*(s;t)$, where

$$r^*(s;t) \quad = \quad r(s;t^*),\ s \in S, t = 1,\cdots,T_1.$$

Since the block bootstrap procedure is performed simultaneously over space, the spatial dependence and space-time dependence (up to temporal lags small compared to B) of the residuals are well kept in the bootstrap sample. In order to mimic the space-time variability of the original high resolution run, we find that it is very important to preserve the space-time dependence of the residuals, which is well captured by simultaneous block bootstrapping. We further remark that our block bootstrap is performed along time, not in space. So this is different from spatial block bootstrap, in which the bootstrap is done over spatial blocks [Lahiri (2003)]. See Section 6 for another approach, where spatial block bootstrap is used.

- Step 3. Synthesize a simulation of a high resolution run by

$$Z_H^*(s;t) = \exp(\hat{a}_t + \hat{b}_t \times \mathrm{BI}[\log\{Z_L(t)\}](s) + r^*(s;t)), \quad 1 \le t \le T_1.$$

- Repeat the above procedure 100 times.

Note that the main randomness is generated in the block bootstrapping of the residuals. A good choice of B is essential to the whole algorithm. If the block size B equals the total number of hours in the simulation, i.e. $B = T_1$, then we get a perfect reproduction of the

original high resolution run with no uncertainty involved, which is obviously not a realistic description of the uncertainty about the high resolution runs given the low resolution runs of the same period. On the other hand, if we let $B = 1$, then the temporal dependence of the residuals is completely lost, which leads to a large bias in terms of the space-time variability. See Section 4 for more discussion. Finally, we remark that this algorithm is guaranteed to yield only positive values.

3.2 Prediction

We simulate $Z_H^{(j)}([T_1+1, T])$ based on $Z_L([1, T_1]), Z_H([1, T_1])$ and $Z_L([T_1+1, T])$. We have the following steps.

- Step 1. Same as Step 1 in the in-sample.
- Step 2. Model $\hat{a}_t, \hat{b}_t, t = 1, \cdots, T_1$ as two univariate time series. Here we apply an AR(2) model, i.e.

$$\begin{aligned} \hat{a}_t &= \mu_a + \beta_{a1}(\hat{a}_{t-1} - \mu_a) + \beta_{a2}(\hat{a}_{t-2} - \mu_a) + \epsilon_{a;t}, \quad t = 3, \cdots, T_1. \\ \hat{b}_t &= \mu_b + \beta_{b1}(\hat{b}_{t-1} - \mu_b) + \beta_{b2}(\hat{b}_{t-2} - \mu_b) + \epsilon_{b;t}, \quad t = 3, \cdots, T_1. \end{aligned}$$

  We obtain the estimates by the conditional least squares method based on the training set, which gives $\hat{\mu}_a = -0.2270, \hat{\beta}_{a1} = 1.3277, \hat{\beta}_{a2} = -0.4448$ and $\hat{\mu}_b = 0.8666, \hat{\beta}_{b1} = 1.3494, \hat{\beta}_{b2} = -0.4563$. Figure 3 shows the auto-correlation and cross-correlation of $\epsilon_{a;t}$ and $\epsilon_{b;t}$ at lags $1 - 40$. The cross correlation of $\epsilon_{a;t}$ and $\epsilon_{b;t}$ at lag 0 is 0.984. It seems that the strong correlation in the original $\hat{a}_t$ and $\hat{b}_t$ (result not shown) is considerably weakened after applying the AR(2) filter above. In the above model, we treat $\epsilon_{a;t}$ and $\epsilon_{b;t}$ as being independent. This simplification seems reasonable in view of the weak cross-correlation in Figure 3. In principle, we could apply a more complex filter to remove the weak correlation at lag 2. However, we shall not complicate our procedure by doing this; this weak correlation will be captured by the block bootstrap (see Step 4 below).
- Step 3. Apply an AR(2) filter to the residuals $r([1, T_1])$, i.e.

$$r(s;t) = \gamma_1 \times r(s;t-1) + \gamma_2 \times r(s;t-2) + e(s;t), \ t = 3, \cdots, T_1.$$

  We get $\hat{\gamma}_1 = 0.9844$ and $\hat{\gamma}_2 = -0.2216$ by pooling all the pixels together in the above least squares regression. Again, the idea of using an AR(2) filter here is to extract out the main dependence in $r(s;t)$ and make the filtered residuals $e(s;t)$ close to independent in time. Since we apply the AR(2) filter in step 2, we also adopt the AR(2) filter for the residuals to make the simultaneous block bootstrap (see Step 4 below) easy to implement.
- Step 4. Simultaneous block bootstrap $\{\epsilon_{a;t}, \epsilon_{b;t}, e(s;t)\}$. Specify the block size B. Again, let $t^*([T_1+1, T])$ be the bootstrapped indexes from the index set $\{1, \cdots, T_1\}$. For $t = T_1 + 1, \cdots, T$, we have $\epsilon^*_{a;t} = \epsilon_{a;t^*}, \ \epsilon^*_{b;t} = \epsilon_{b;t^*}, \ e^*(s;t) = e(s;t^*)$.

- Step 5. Set $a^*_{T_1} = \hat{a}_{T_1}$, $a^*_{T_1-1} = \hat{a}_{T_1-1}$, $b^*_{T_1} = \hat{b}_{T_1}$ and $b^*_{T_1-1} = \hat{b}_{T_1-1}$. Similarly, let $r^*(s;T_1) = r(s;T_1)$ and $r^*(s;T_1-1) = r(s;T_1-1)$. We sequentially obtain the bootstrapped residuals $r^*(s;t)$, $s \in S$ and predict a^*_t and b^*_t, $t = T_1+1, \cdots, T$, i.e.

$$\begin{aligned} a^*_t &= \hat{\mu}_a + \hat{\beta}_{a1}(a^*_{t-1} - \hat{\mu}_a) + \hat{\beta}_{a2}(a^*_{t-2} - \hat{\mu}_a) + \epsilon^*_{a;t}, \\ b^*_t &= \hat{\mu}_b + \hat{\beta}_{b1}(b^*_{t-1} - \hat{\mu}_b) + \hat{\beta}_{b2}(b^*_{t-2} - \hat{\mu}_b) + \epsilon^*_{b;t}, \\ r^*(s;t) &= r^*(s;t-1) \times \hat{\gamma}_1 + r^*(s;t-2) \times \hat{\gamma}_2 + e^*(s;t). \end{aligned}$$

  Note that the bootstrap is performed simultaneously over a_t,b_t and $r(s;t)$ to capture the dependence among them. For example, the dependence of a_t and $r(s;t)$ is well kept by this procedure while it is lost if we do not do the bootstrap simultaneously. Here the AR(2) filtering is combined with block bootstrap to better preserve the dependence and also make the choice of block size less critical. In the literature of resampling, this procedure is called post-blackening [Davison and Hinkley (1997)]. It has been applied by Srinivas and Srinivasan (2005) in the stochastic simulation of multi-site multi-season streamflows for a similar purpose. Note that the spatial dependence and temporal dependence (at small lags) of $r(s;t)$ are well captured by the post-blackening procedure since our block bootstrap is performed simultaneously over space.

- Step 6. Synthesize a simulation of a high resolution run by

$$Z^*_H(s;t) = \exp(a^*_t + b^*_t \times \mathrm{BI}[\log\{Z_L(t)\}](s) + r^*(s;t)),\ t = T_1+1, \cdots, T.$$

- Repeat the above procedure 100 times.

4 Assessing Simulations

We first describe some verification measures used in this paper, which apply to both in-sample and prediction. Given the ensemble of high resolution simulations, one can in fact look at many summary statistics. Since our main interest is to see if our simulations preserve the space-time variability, we focus on the following measures. The block size is taken to be 16 for both in-sample and prediction in the results reported here.

For a space-time process $Z(s;t)$, $s \in S$ stands for spatial location and $t = 1, \cdots, T$ stands for time. Given spatial lag h and time lag l, we define the space-time (empirical) variogram $V(h,l)$ as

$$V(h,l) = \frac{1}{2N(h,l)} \sum \{Z(s;t) - Z(s+h;t+l)\}^2,$$

where the sum is over all (s,t) for which $\{Z(s;t), Z(s+h;t+l)\}$ is observed and $N(h,l)$ is the number of such (s,t). In our case, $h = 1$ means 2km apart and $l = 1$ means 1 hour apart. This is a natural measure of the variability in both space and time. At temporal lag zero, $V(h,0)$ is the temporal average of the spatial variogram at spatial lag h. Empirically, we find that the

original and simulated space-time field is fairly isotropic in space so for each temporal lag we only show the space-time variogram along the north-south direction. Taking the average over all directions does not change the results much.

Figure 4 shows the space-time variogram of the high resolution runs in the training set in comparison with the space-time variogram based on 100 simulations for $B = 16$ and $B = 1$. For $B = 16$, the true variogram line lies well within the 90% simulation band. In contrast, an apparent bias occurs for $B = 1$ at nonzero temporal lags. The bias is large, especially at small spatial lags. In addition, note that the variability in the simulated variograms is smaller for $B = 1$ than that for $B = 16$. Figure 5 shows the space-time variogram for the prediction. The apparent bias could be due to the prediction error of $a^*[T_1 + 1, T]$ and $b^*[T_1 + 1, T]$, and different stochastic behavior of residuals in $[T_1 + 1, T]$. That is, the residuals for the training set do not quite match the statistical features of residuals in the testing set. We will leave further explanations to the next section.

We also look at the temporal average of the spatial variogram of first differences in time [Stein (2005)]. In other words, we first get $\Delta Z(s;t) = Z(s;t+1) - Z(s;t), t = 1, \cdots, T_1 - 1$, then compute the spatial variograms of $\Delta Z([1, T_1 - 1])$ averaged over time. Figure 6a shows that the truth is well included in the 90% simulation band for the in-sample, while Figure 6b indicates that the truth is overestimated for the prediction. Figures 7a and 7b show $V(0, l)$, $l = 1, \cdots, 10$ for the in-sample and prediction respectively. Again, the true variogram is well within the 90% simulation band for the in-sample and a bias arises for the prediction.

For a stationary time series, the Fourier spectrum is a powerful tool to look at the distribution of the total variability over frequencies. Since each pixel in space is associated with a time series, we look at the average of the log-spectra over pixels. Figures 8a and 8b show the spatial average of $\log_{10}$ of the empirical spectra for the in-sample and prediction. The true spectrum line stays well inside the 90% simulation band for most frequencies in the case of in-sample, while the truth is a bit out of the band for the prediction, especially at low and medium frequencies. In the in-sample, it seems that the $\log_{10}$ of the spectra at the daily frequency $\pi/12$ is well within the simulation band, which suggests that taking block size 16 does not produce much bias in capturing the diurnal cycles.

In ensemble weather forecasting, the rank histogram has been used as a verification criteria [cf. Hamill and Colucci (1997) and Gel et al. (2004)]. Since our problem bears some resemblance to probabilistic weather forecasting, we provide some results for rank histograms here. To obtain the rank histogram, we simply calculate the rank of $Z_H(s;t)$ when pooled with 100 simulations $Z_H^{(j)}(s;t)$, $j = 1, \cdots, 100$. Then we pool all the ranks over space and time and plot the histogram of the ranks. Here a rank of 100 indicates that all the simulated values are higher than the true value. Note that for two neighboring pixels or one pixel at consecutive times, we cannot expect their ranks to be nearly independent. Nevertheless, the rank histogram provides a meaningful way to evaluate our simulated high resolution outputs. If the simulations were capturing the stochastic behavior of the original space-time field, the rank histogram should be

close to uniform. Systematic deviations from uniformity indicate problems. For example, a U-shaped rank histogram results if the simulated processes are less variable (in terms of marginal variability) than the actual process [cf. Hamill (2001)]. Figures 9a and 9b show the rank histograms for the in-sample and prediction respectively. The histogram looks quite uniform for the in-sample and a bit worse for the prediction. This again suggests that our simulation algorithm performs well for the in-sample and an apparent bias shows up in the prediction. Further discussion can be found in Section 6.

5 Frequency domain regression

As we see from Figure 2, the coherence between the bilinear interpolation of low resolution output and the residuals is strengthened at high frequencies, which seems to suggest that the time domain regression cannot fully decorrelate these two. A more effective way of decorrelation can be performed in the frequency domain.

1. First transform the high resolution output and bilinear interpolation of low resolution output into frequency domain, i.e. for each pixel, transform the associated time series into Fourier coefficients. Let $\lambda_k = 2\pi k/T_1, k = 0, \cdots, T_1 - 1$. For fixed s, we have

$$\begin{aligned} \{\log(Z_H(s;t)), t = 1, \cdots, T_1\} &\stackrel{F}{\Rightarrow} \{W_H(s;\lambda_k), k = 0, \cdots, T_1 - 1\} \\ \{\mathrm{BI}[\log\{Z_L(t)\}](s), t = 1, \cdots, T_1\} &\stackrel{F}{\Rightarrow} \{W_L(s;\lambda_k), k = 0, \cdots, T_1 - 1\}, \end{aligned}$$

   where "$\stackrel{F}{\Rightarrow}$" stands for the Fourier transform, for example,

$$W_H(s;\lambda_k) = \frac{1}{\sqrt{T_1}} \sum_{t=1}^{T_1} \log(Z_H(s;t)) e^{\sqrt{-1}t\lambda_k}, \ k = 0, \cdots, T_1 - 1.$$

2. Perform the regression in the spectral domain for each frequency, i.e.

$$W_H(s;\lambda_k) = \beta_k W_L(s;\lambda_k) + R(s;\lambda_k), s \in S. \tag{1}$$

   Find the least squares estimate of β_k and denote it by $\hat{\beta}_k$. For each k, $\hat{\beta}_k$ is a complex number. Note that the least square norm is minimized in the complex case.

3. Transform $R(s;\lambda_k)$ back to the time domain for each s:

$$R(s;\lambda_k) \stackrel{F^{-1}}{\Rightarrow} E(s;t);$$

   where $\stackrel{F^{-1}}{\Rightarrow}$ stands for the inverse Fourier transform.

4. Perform simultaneous bootstrap for $E(s;t)$, i.e. $E^*(s;t) = E(s;t^*)$, where $t^*([1,T_1])$ are bootstrapped indexes with block size 16. Then transform $E^*(s;t)$ into the Fourier domain, i.e. $E^*(s;t) \stackrel{F}{\Rightarrow} R^*(s;\lambda_k)$.

5. Plug the bootstrapped residuals into the frequency domain regression equation,

$$W_H^*(s;\lambda_k) = \hat{\beta}_k W_L(s;\lambda_k) + R^*(s;\lambda_k), s \in S.$$

6. Take the inverse Fourier transform of $W_H^*(s;\lambda_k)$ and then exponentiate pointwise to go back to the original scale.

$$\{W_H^*(s;\lambda_k), k = 0, \cdots, T_1 - 1\} \overset{F^{-1}}{\Rightarrow} \{\log Z_H^*(s;t), t = 1, \cdots, T_1\}.$$

Then $Z_H^*(s;t) \Leftarrow \exp(\log Z_H^*(s;t))$.

As indicated in Figure 2, the coherence of the bilinear interpolation of low resolution output with the residuals obtained by frequency domain regression is exactly zero. This is implied by the regression (1) in the frequency domain, which makes $W_L(s;\lambda_k)$ orthogonal to $R(s;\lambda_k)$ over space for each k. Figure 4 shows the space-time variogram at temporal lags 0-3. The true variogram line is very close to the 95% percentiles based on 100 simulations. Figures 6a and 7a show spatial variograms of first difference in time and spatial average of temporal variograms respectively. Both indicate some bias. Figures 8c and 9a show spatial average of $\log_{10}$ spectrum and rank histogram, which suggest that the bias is not large. Overall, the frequency domain regression method seems to capture the conditional distribution to some extent, but Figures 6a, 7a and Figure 4 suggest that the 100 simulations are slightly underestimating the space-time variability of the true high resolution output.

We can adapt the above algorithm from in-sample to prediction. Here we assume $T - T_1 = T_1 = 288$, so the least square estimates $\hat{\beta}([0, T_1 - 1])$ for the training set can be directly used for the testing set. In addition, the prediction can be done with the residuals for the testing set bootstrapped (time domain bootstrap) directly from the training set. Figure 5 shows that the space-time variograms at temporal lags 0-3 are inside the 90% simulation bands. Figures 6b, 7b, 8d show the spatial variograms of first difference in time, spatial average of temporal variogram, and spatial average of $\log_{10}$ empirical spectrum. All seem to suggest that the 100 simulations capture the space-time variability of the truth pretty well, with an indication of slight underestimation of the variability. Figure 9b shows a U-shaped rank histogram, which suggests an underestimation of marginal variability. We also looked at the pixelwise mean square error of the average of 100 simulations to the truth for the testing set. The mean square error is 12,848 for frequency domain regression and 14,470 for time domain regression, both of which are smaller than 18,522, the pixelwise mean square error of the bilinear interpolation of low resolution output to the high resolution run for the testing set. This is partly expected since we borrow the high resolution information from the training period in our simulations. Overall, the prediction by frequency domain regression does a better job than the prediction by time domain regression.

In the case where $T_1 \neq T - T_1$, i.e. the training and testing sets have different time length, then one can do interpolation of $\hat{\beta}([0, T_1 - 1])$ over frequencies to get the corresponding coefficients for the testing set. Again the residuals can be bootstrapped from the training set.

6 Discussion

To summarize, the idea of our conditional simulation is to filter out the component (the residuals) in high resolution runs that we might expect are approximately independent of the low resolution runs and perform a simultaneous block bootstrap of the residuals. Taking the time domain regression procedure for example, let f denote the nonlinear operations involved in the filtering step. Hence we have

$$Z_H([1,T_1]) = f(Z_L([1,T_1]), \{\hat{a}_t, \hat{b}_t, r(t), 1 \le t \le T_1\}). \tag{2}$$

Define $r^*(t) = \{r^*(s;t), s \in S\}$. Then for the in-sample, we write

$$Z_H^*([1,T_1]) = f(Z_L([1,T_1]), \{\hat{a}_t, \hat{b}_t, r^*(t), 1 \le t \le T_1\}),$$

where $r^*([1,T_1])$ can be regarded as a perturbation of the residuals $r([1,T_1])$. Consequently, we can interpret $Z_H^*([1,T_1])$ as the perturbed output with the nonlinear structural constraint in (2).

Similarly, in the prediction procedure, we have

$$Z_H^*([T_1+1,T]) = f(Z_L([T_1+1,T]), \{a_t^*, b_t^*, r^*(t), T_1+1 \le t \le T\}),$$

where a_t^* and b_t^* are predicted intercept and slope, $r^*([T_1+1,T])$ are the post-blackened residuals from the training set. The prediction bias we see in the various verification measures could be due to the fact that the predicted residuals $r^*(t)$ from the training set are not sufficiently stochastically similar to the true residuals for the testing set. In addition, the predicted a_t^* and b_t^* do not quite match the true values for the testing set. Figure 10 shows the mean, 5th percentile and 95th percentile of $b^*([T_1+1,T])$ based on 100 simulations. It seems that the mean of 100 simulations deviates from the mean of the true slope over time quite a bit, although the 90% simulation band covers the true slope most of the time. Qualitatively similar patterns can be found for the predicted intercept $a^*([T_1+1,T])$. To study the effect of using estimates of a_t and b_t based on the training data, we pretend we know the true intercept and slope for the testing set and still use the post-blackened residuals from the training set. The bias is reduced but still exists. We attribute this to nonstationarity of the residuals, although it is difficult to identify this nonstationarity explicitly.

To highlight the importance of reproducing the overall space-time variability, we investigate the ability of our simulation scheme to capture the extremes in the in-sample step. For each hour, we get a maximum over space and calculate its rank when pooled with 500 simulated maxima (i.e. each one is the maximum over space for a particular simulation). We pool all the ranks together (over hours) and plot its histogram. Again, the neighboring ranks are not independent; nevertheless, this provides a sensible way to see if our simulation can capture the stochastic features of spatial extremes. Here 500 simulations were performed to minimize the

sampling variability in plotting the rank histograms. The results do not change much if we increase the simulation times to 1000 or more. Figures 11a and 11b show the rank histograms for the time domain regression and frequency domain regression respectively. Both histograms look quite uniform, which suggests that both simulation schemes capture the stochastic behavior of spatial extremes fairly well. We also plot the rank histogram for the 4-hour spatial maximum. That is, for every nonoverlapping 4-hour period (for example, Aug 1st, 1am-4am, 5am-8am etc.), we get its maximum over space and 4-hour period and compare with the corresponding values from 500 simulations. Figures 11d and 11e show the rank histograms of 4-hour spatial maxima for the time domain regression and frequency domain regression. Figure 11e looks more uniform than Figure 11d, which suggests that time domain regression produces more bias in terms of space-time extremes. A similar pattern was found for the spatial maximum over a longer period (say 8 hours).

We also tried a "naive approach" that partly ignores the spatial and temporal variability in producing the high resolution run based on the runs we have. Let $S_1 = \{1, \cdots, 96\}$ and $S_2 = \{1, \cdots, 112\}$ be index sets along the N-S and W-E directions respectively. We describe it as follows.

- Form the ratios of the point value in the high resolution run with its corresponding value in the low resolution, i.e.

$$R(i_1, i_2; t) = \frac{Z_H(i_1, i_2; t)}{Z_L(\lfloor (i_1 - 1)/4 \rfloor + 1, \lfloor (i_2 - 1)/4 \rfloor + 1; t)}, \ i_1 \in S_1, i_2 \in S_2, 1 \le t \le T_1.$$

  Here $\lfloor a \rfloor$ denotes the integer part of a.

- Block bootstrap the ratios $R(i_1, i_2; t)$ spatially independently from time to time. Here each block in space is of size 16×16. Let $\{\tilde{R}(k, l; t), k = 1, \cdots, 6, l = 1, \cdots, 7\}$ denote the 6×7 blocks at time t. For example, $\tilde{R}(1, 1; t) = \{R(i_1, i_2; t), i_1 = 1, \cdots, 16, i_2 = 1, \cdots, 16\}$. Then for the bootstrapped ratio $R^*(i_1, i_2; t)$, its corresponding block version is obtained by

$$\tilde{R}^*(k, l; t) = \tilde{R}(J_k, H_l; t), \ k = 1, \cdots, 6, \ l = 1, \cdots, 7,$$

  where $J_k, \ k = 1, \cdots, 6$ and $H_l, \ l = 1, \cdots, 7$ are independent and identically distributed random draws from the sets $\{1, \cdots, 6\}$ and $\{1, \cdots, 7\}$ respectively.

- Synthesize the high resolution runs by multiplying the bootstrapped ratio matrix by the corresponding values from the low resolution runs, i.e. for $i_1 \in S_1, i_2 \in S_2, 1 \le t \le T_1$,

$$Z_H^*(i_1, i_2; t) = R^*(i_1, i_2; t) \times Z_L(\lfloor (i_1 - 1)/4 \rfloor + 1, \lfloor (i_2 - 1)/4 \rfloor + 1; t).$$

- Repeat the above procedure 500 times.

This approach is in spirit similar to the one taken by Herwehe et al. (2004), who fit a Weibull distribution to the values from the high resolution run for each low resolution grid cell and fit parameters of the distribution independently from grid cell to grid cell over time. The above procedure maintains the dependence within each block but ignores the dependence across blocks and over time. Figures 11c and 11f show the rank histograms of hourly spatial maxima and 4-hour spatial maxima. Figure 11c is not as uniform as Figures 11a and 11b, while Figure 11f shows a sharp peak on the right, which suggests that the 4-hour spatial maximum is overestimated. This phenomenon is presumably due to the loss of time dependence in this naive procedure. Taking spatial block size less than 16 (for example, 4) produces less uniform rank histograms with a sharper peak on the right (result not shown). In comparison with the procedure we proposed in Sections 3 and 5, this naive approach is inferior. From this example, we see a close relationship between the space-time variability and the space-time extremes. This further justifies our goal of reproducing the overall space-time variability. Although there is a risk of overinterpreting these rank histograms, we believe that a sensible approach to sub-grid variability should take into account the overall space-time variability. From a statistical point of view, any statistic (for example, space-time extreme) is a function of the joint distribution of the whole space-time field. A sensible way of addressing the sub-grid variability needs to capture this joint distribution at least approximately.

7 Conclusions

Our approach is conceptually close to the geostatistical output perturbation method adopted by Gel et al. (2004) to produce calibrated probabilistic weather forecasts. However, there are essential differences between the two. First of all, our algorithm is proposed under a space-time framework, whereas their treatment is limited to the purely spatial setting. Our perturbation is to the residuals, which is obtained by nonlinearly filtering our high resolution runs given corresponding low resolution runs. Their perturbation is to perturb the deterministic forecast (output) under the assumption of an additive model structure. Their procedure involves the modeling, parameter estimation as well as the simulation of a Gaussian random field and hence is more complex in its practical implementation. In contrast, our algorithm is quite simple and can be readily used in a variety of settings.

A possible way to reduce the bias we get in prediction from 8km to 2km is to extrapolate the bias in the prediction from 32km to 8km resolution. The details can be found in Shao and Stein (2005). We also tried other species, such as O_3 and NO_2. Using exactly the same simulation scheme, we find that the results are not as good as for CO for both in-sample and prediction. The main reason is that the simple nonlinear filter we use here is no longer appropriate for these photochemically active species. However, the idea of nonlinear filtering with perturbation of the residuals still applies, although it is difficult to find a good nonlinear filter.

As mentioned earlier, the choice of block size B controls the magnitude of bias and variance. We do not try to come up with an optimal B here, since that would involve lengthy computations and, in any case the best B will depend on the specific criteria. For the purpose of conditional simulation, one can always try different block sizes and choose an appropriate one. We tried $B = 12, 16, 24$ and quantitatively similar results were obtained. A rough guideline is that B should be large enough to preserve the weak space-time dependence within the residuals. We also tried $B = 96$, which produced a similar result as $B = 16$ with no substantial improvement. We further tried different sizes for the training set and testing set. For example, we tried $T_1 = 24 \times 16$, i.e. the first 16 days as the training set and last 8 days as the testing set. The results we obtained were qualitatively similar to what we present here.

In conclusion, we propose a novel statistical approach to representing the sub-grid variability while maintaining the overall space-time variability. Additionally, we provide a set of statistical tools for the practitioner's use to summarize sub-grid variability. In the situation where one has low resolution runs for a long period and only a few days' high resolution runs, the prediction algorithm proposed here can be used to generate random high resolution runs as a surrogate to the real high resolution runs, which are very computationally expensive to get. Since CMAQ considers multi-species' interactions, it would be interesting to study how to apply this idea to conditionally simulate more than one species at high resolution simultaneously. Furthermore, it would also be interesting to see how the output of a human exposure model changes as we change the sub-grid variability information.

8 Appendix

For two space-time fields $Z_1(s;t)$ and $Z_2(s;t)$, $s \in S, 1 \le t \le T_1$, we first define the pixelwise periodogram and cross-periodogram by

$$\begin{aligned} I_{11}(s;\lambda_k) &= \frac{1}{2\pi T_1}\left|\sum_{t=1}^{T_1} Z_1(s;t)e^{it\lambda_k}\right|^2, \quad \lambda_k = 2\pi k/T_1, k = 0, 1, \cdots, T_1 - 1. \\ I_{22}(s;\lambda_k) &= \frac{1}{2\pi T_1}\left|\sum_{t=1}^{T_1} Z_2(s;t)e^{it\lambda_k}\right|^2, \\ I_{12}(s;\lambda_k) &= \frac{1}{2\pi T_1}\sum_{t=1}^{T_1} Z_1(s;t)e^{it\lambda_k} \times \sum_{t=1}^{T_1} Z_2(s;t)e^{-it\lambda_k}. \end{aligned}$$

For $0 \le k \le T_1 - 1$, we define the average coherence between Z_1 and Z_2 at frequency λ_k by

$$\text{Avecoh}(Z_1, Z_2; \lambda_k) := \frac{|\sum_{s\in S} I_{12}(s;\lambda_k)|}{\sqrt{\sum_{s\in S} I_{11}(s;\lambda_k)\sum_{s\in S} I_{22}(s;\lambda_k)}}.$$

9 Acknowledgements

The authors gratefully acknowledge Robin Dennis for preparing the CMAQ runs used in this paper. We also want to thank Jason Ching and Jenise Swall for helpful comments on an earlier version. The first author thanks Jason Ching for his hospitality during a visit to EPA.

10 References

Byun, D. W., and J. K. S. Ching, Science algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System, 1999. Available at

http://www.epa.gov/asmdnerl/CMAQ/CMAQscienceDoc.html

Chan, S. J., J. C. Liljegren, W. J. Shaw, J. H. Hubbe, and J. C. Doran, Influence of sub-grid variability on surface hydrology. *J. Climate.*, 10, 3157-3166, 1997.

Chilès, J., and P. Delfiner, *Geostatistics: Modeling Spatial Uncertainty.* New York: Wiley, 1999.

Ching, J. K. S., T. Pierce, T. Palma, W. Hutzell, R. Tang, A. Cimorelli, and J. Herwehe, Linking air toxic concentrations from CMAQ to the HAPEM-5 exposure model at neighborhood scales for the Philadelphia area. From the 16th conference of Biometeorology and Aerobiology, Vancouver, BC, Canada, Amer. Meteor. Soc. 2004.

Ching, J. K. S., J. Herwehe, and J. Swall, On joint deterministic grid modeling and sub-grid variability conceptual framework for model evaluation. Preprint, 2005.

Davison, A. C., and D. V. Hinkley, *Bootstrap methods and their application.* Cambridge University Press, 1997.

Gel, Y., A. E. Raftery, and T. Gneiting, Calibrated Probabilistic mesoscale weather field forecasting: the geostatistical output perturbation method. *J. Am. Stat. Assoc.*, 99(467), 575-590, 2004.

Hamill, T. M., Interpretation of rank histograms for verifying ensemble forecasts. *Mon. Weather Rev.*, 129(3), 550-560, 2001.

Hamill, T. M., and S. J. Colucci, Verification of Eta-RSM short-range ensemble forecasts. *Mon. Weather Rev.*, 125, 1312-1327, 1997.

Harris D., and E. Foufoula-Georgiou, Sub-grid variability and stochastic downscaling of modeled clouds: Effects on radiative transfer computations for rainfall retrieval. *J. Geophys. Res.*, 106(D10), 10349-10362, 2001.

Herwehe, J., J. K. S. Ching, and J. Swall, Quantifying sub-grid pollutant variability in eulerian air quality models. In 13th Joint Conference on the Applications of Air Pollution Meteorology with the Air and Waste Management Association, 23-27 August 2004. Vancouver, BC, Canada, American Meteorological Society. ATDD Contribution File No. 04/23.

Künsch, H. R., The jackknife and the bootstrap for general stationary observations, *Ann. Statist.*, 17(3), 1217-1241, 1989.

Lahiri, S. N., *Resampling methods for dependent data*. Springer-Verlag, 2003.

Nunes, C., and A. Soares, Geostatistical space-time simulation model for air quality prediction. *Environmetrics*, 16, 393-404, 2005.

Perica, S., and E. Foufoula-Georgiou, A model for multiscale disaggregation of spatial rainfall based on coupling meteorological and scaling descriptions, *J. Geophys. Res.*, 101(D21), 26347-26361, 1996.

Shao, X., and M. L. Stein (2005), Statistical conditional simulation of a multiresolution numerical air quality model. CISES Technical Report 31, University of Chicago. Available at

http://www.stat.uchicago.edu/~cises/research/cises-tr31.pdf

Shao, X., M. L. Stein, and J. K. S. Ching, Statistical comparisons of methods for interpolating the output of a numerical air quality model. CISES Technical Report 23, University of Chicago, 2005. Available at

http://www.stat.uchicago.edu/~cises/research/cises-tr23.pdf

Stein, M. L., Space-time covariance functions, *J. Am. Stat. Assoc.*, 100(469), 310-321, 2005.

Srinivas, V. V., and K. Srinivasan, Hybrid moving block bootstrap for stochastic simulation of multi-site multi-season streamflows. *J. Hydrol.*, 302, 307-330, 2005.

Venugopal, V., E. Foufoula-Georgiou, and V. Sapozhnikov, A space-time downscaling model for rainfall, *J. Geophys. Res.*, 104 (D16), 19705-19721, 1999.

Vogel, R. M., and A. L. Shallcross, The moving blocks bootstrap versus parametric time series models. *Water Resour. Res.*, 32(6), 1875-1882, 1996.

Wood, A. W., L. R. Leung, V. Sridhar and D. P. Lettenmaier, Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs. *Climatic Change* 62(1-3), 189-216, 2004.

Figure 1: The picture on the left is the 2km resolution CO at 1am on Aug 1st, 1999. The one on the right hand side corresponds to the 8km resolution CO at 1am on Aug 1st, 1999. The unit is ppmV.

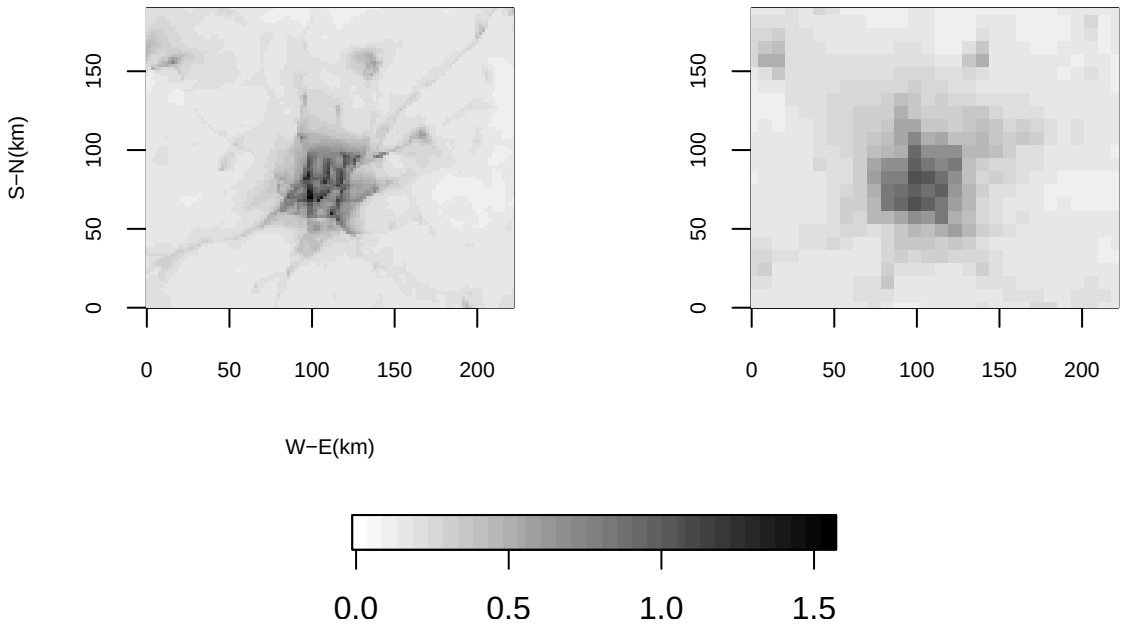

Figure 2: Average coherences: The dashed (thin) line indicates average coherence between the high resolution output and bilinear interpolation of low resolution output (at log scale) for the training set. The solid (gray) line indicates average coherence between the residuals (time domain regression) and the bilinear interpolation of low resolution output (at log scale) for the training set. For frequency domain regression, average coherence between the residuals and the bilinear interpolation of low resolution output (at log scale) for the training set are exactly zero, as indicated by the solid black line.

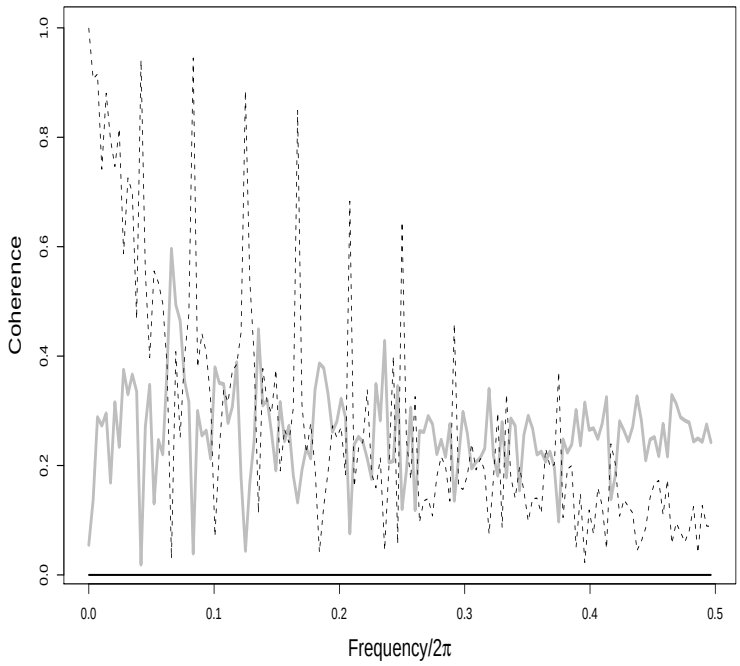

Figure 3: The auto-correlation and cross-correlation of $\{\epsilon_{a;t}, \epsilon_{b;t}, t = 3, \cdots, T_1\}$ at lags 1 through 40.

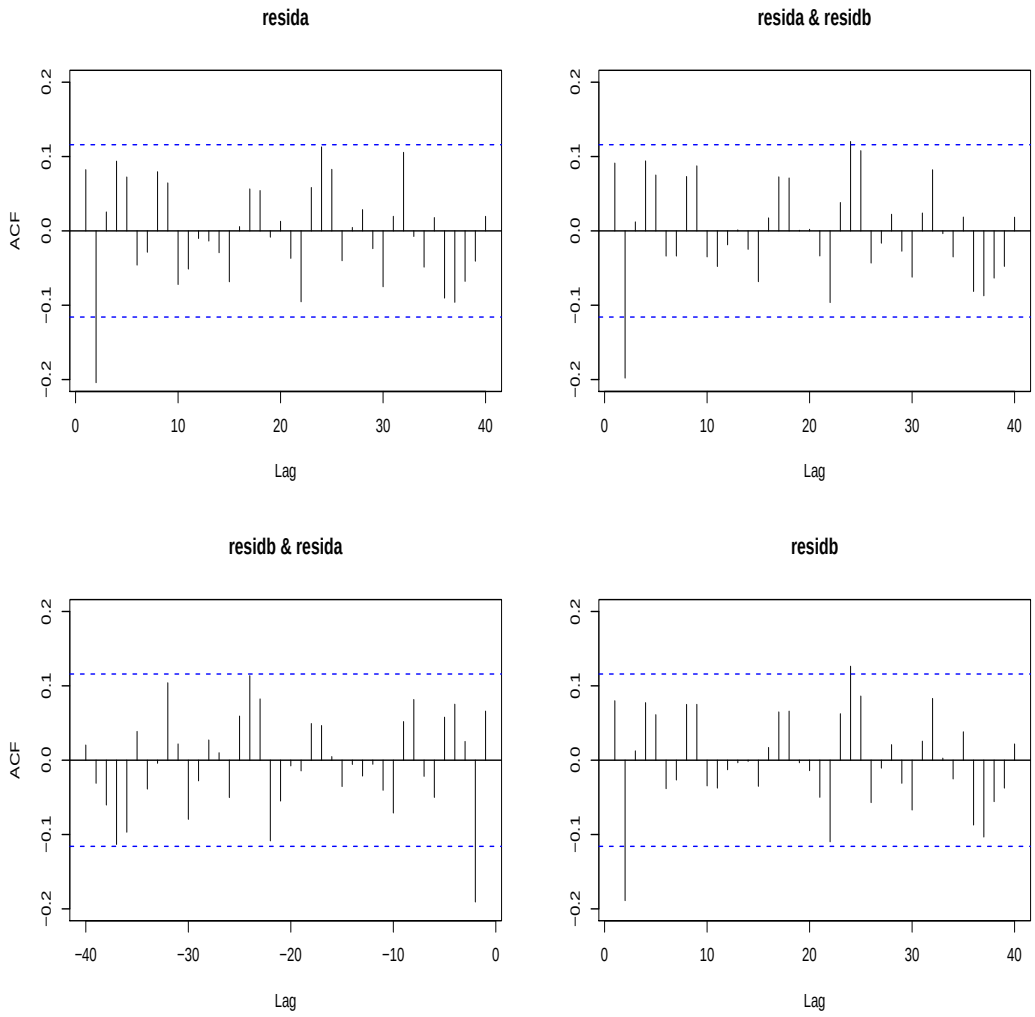

Figure 4: In-sample: The space-time variogram of $Z_H([1, T_1])$ and $Z_H^{(j)}([1, T_1])$, $j = 1, \cdots, 100$ at temporal lags 0-3. For each spatial and temporal lag, quantiles (5%, 95%) of 100 simulations are displayed in bars.

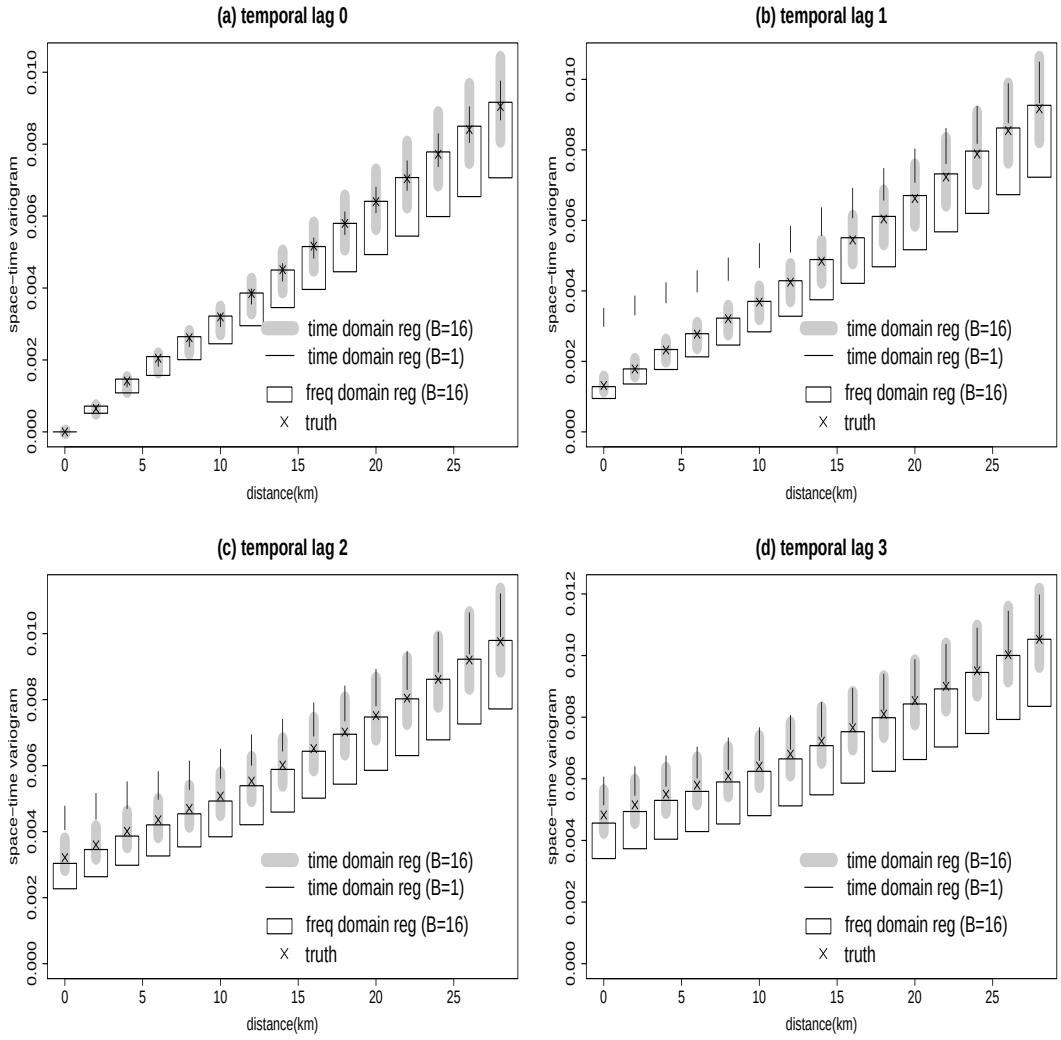

Figure 5: Prediction: The space-time variogram of $Z_H([T_1+1,T])$ and $Z_H^{(j)}([T_1+1,T])$, $j = 1,\cdots,100$ at temporal lags 0-3. The format is same as Figure 4.

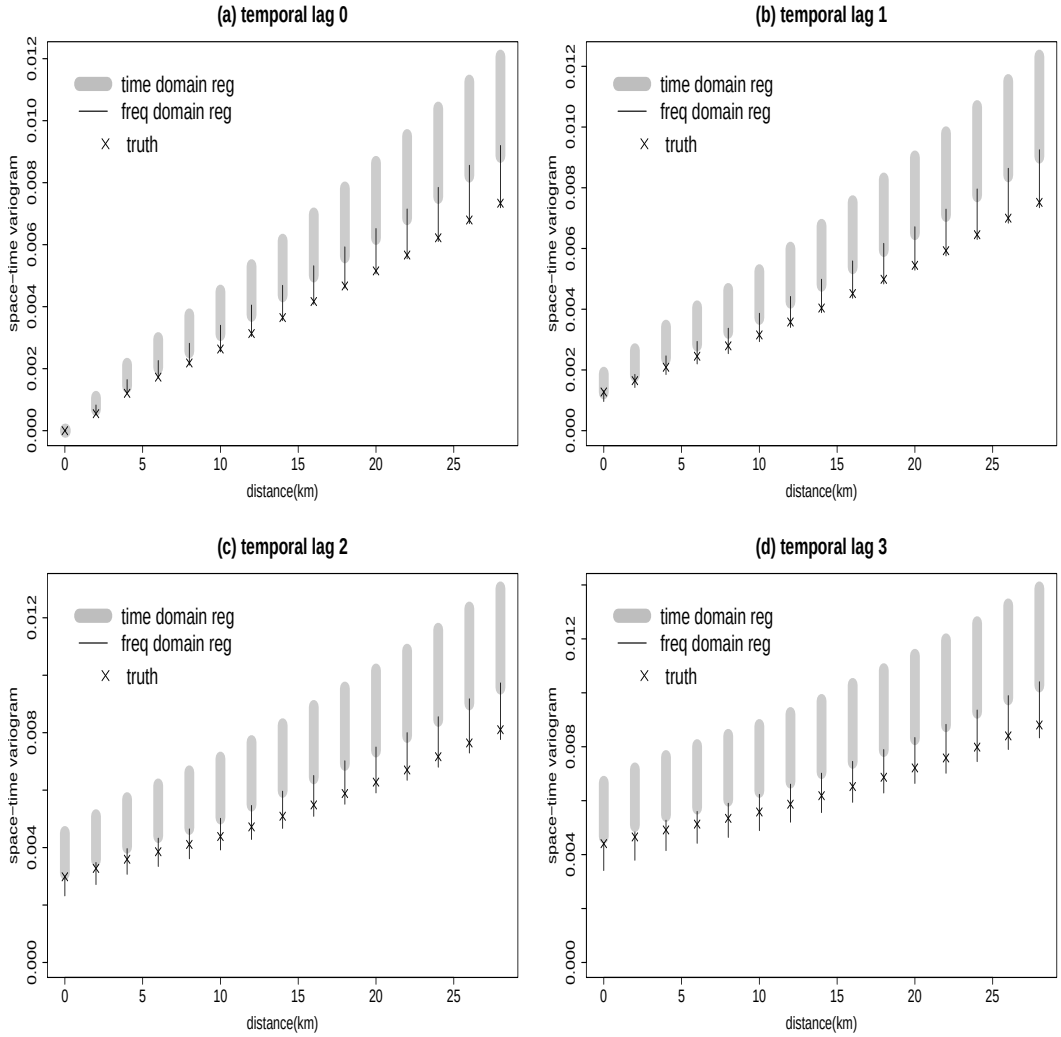

Figure 6: Spatial variograms of first difference in time. (a). In-sample by time domain regression and frequency domain regression. (b). Prediction by time domain regression and frequency domain regression. For each spatial lag, quantiles (5%, 95%) of 100 simulations are displayed in bars.

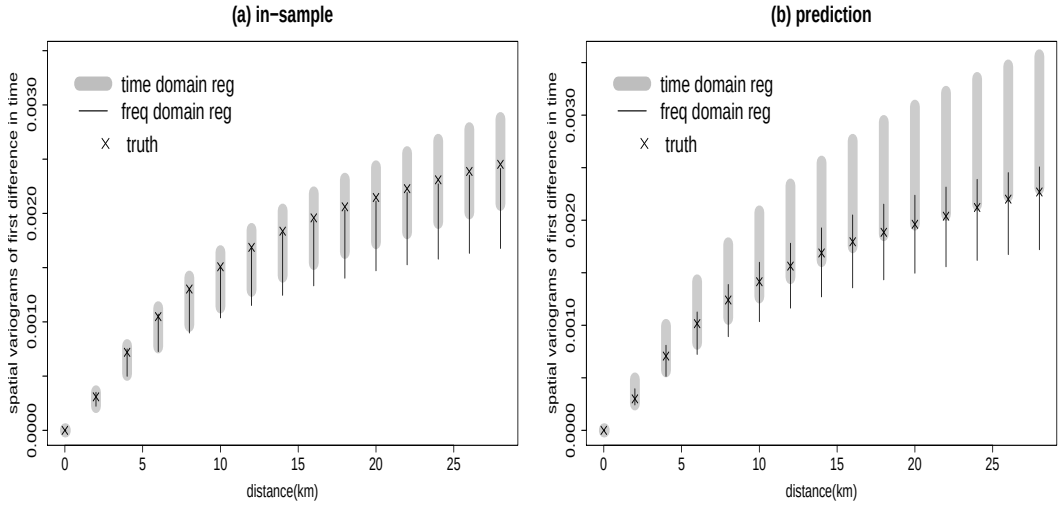

Figure 7: Spatial average of temporal variogram. (a). In-sample by time domain regression and frequency domain regression. (b). Prediction by time domain regression and frequency domain regression. For each temporal lag, quantiles (5%, 95%) of 100 simulations are displayed in bars.

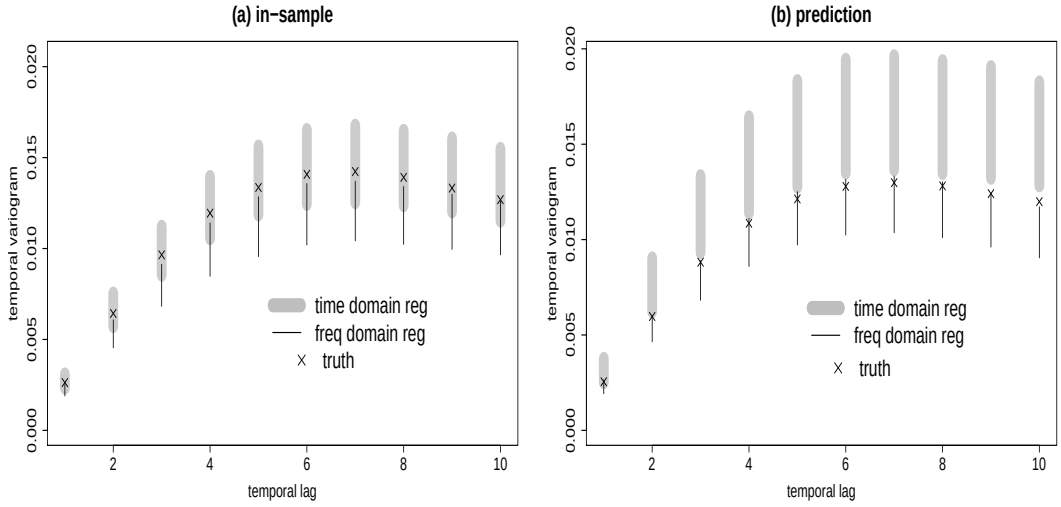

Figure 8: Spatial average of $\log_{10}$ empirical spectrum. The empirical spectrum for each pixel is calculated using the command "spectrum" in R with spans=(5,7). (a) The in-sample by time domain regression. (b). The prediction by time domain regression. (c). The in-sample by frequency domain regression. (d). The prediction by frequency domain regression. The stripe in the following plots is the 90% pointwise simulation band. Note that the frequency in the horizontal axis is the real frequency divided by 2π.

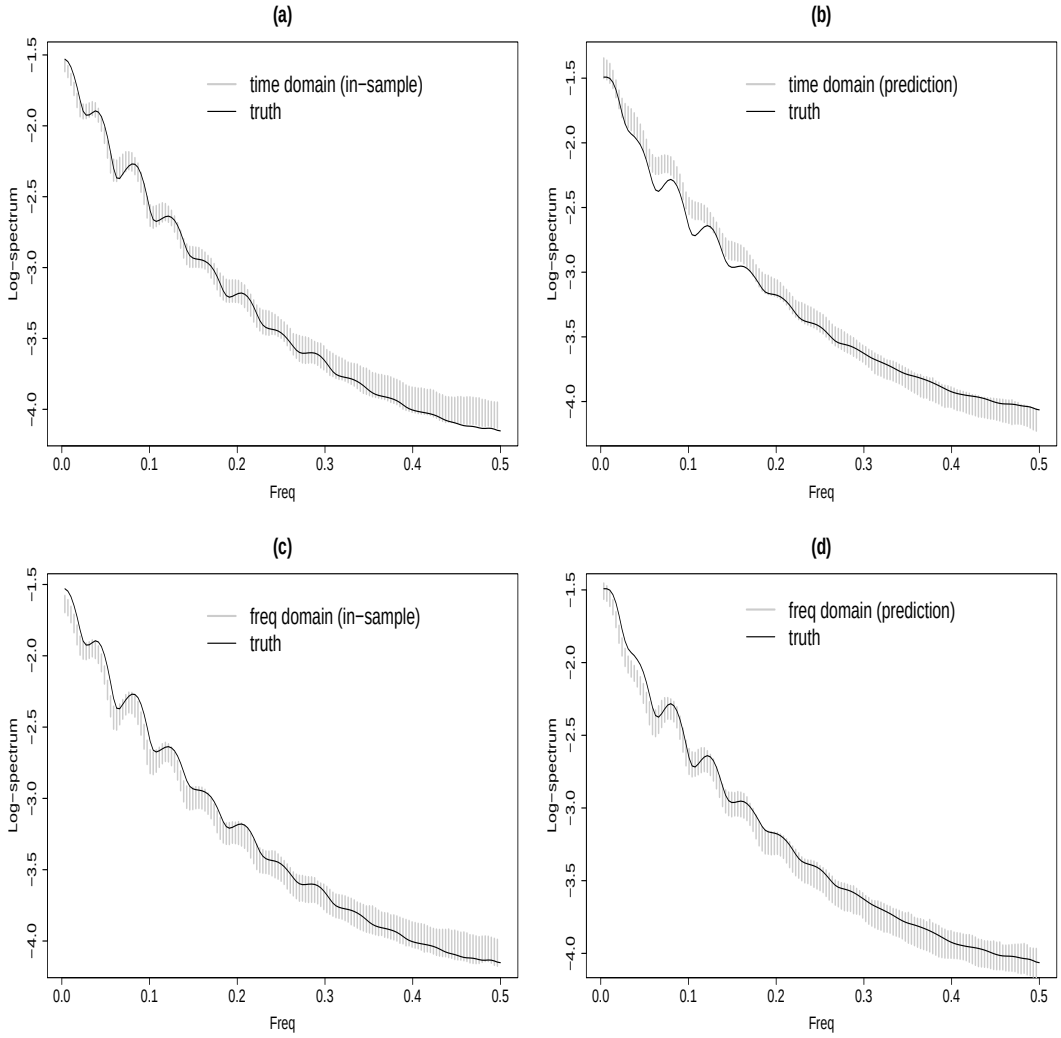

Figure 9: Rank histogram. (a). The gray region corresponds to the in-sample by time domain regression, while the overlaid transparent one corresponds to the in-sample by frequency domain regression. (b). The gray region corresponds to the prediction by frequency domain regression and the overlaid transparent one corresponds to the prediction by time domain regression.

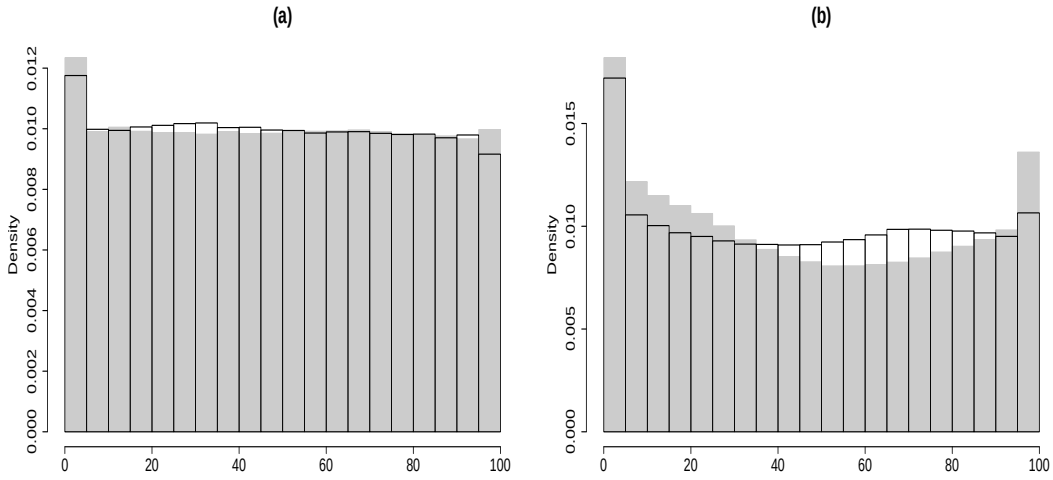

Figure 10: For each hour in the testing set, i.e. $t = T_1 + 1, \cdots, T$, we show the mean ("+"), 5th percentile ("$\circ$") and 95th percentile ("$\triangle$") of the predicted slopes based on 100 simulations along with the true slope (solid line) and its mean over time (dashed line).

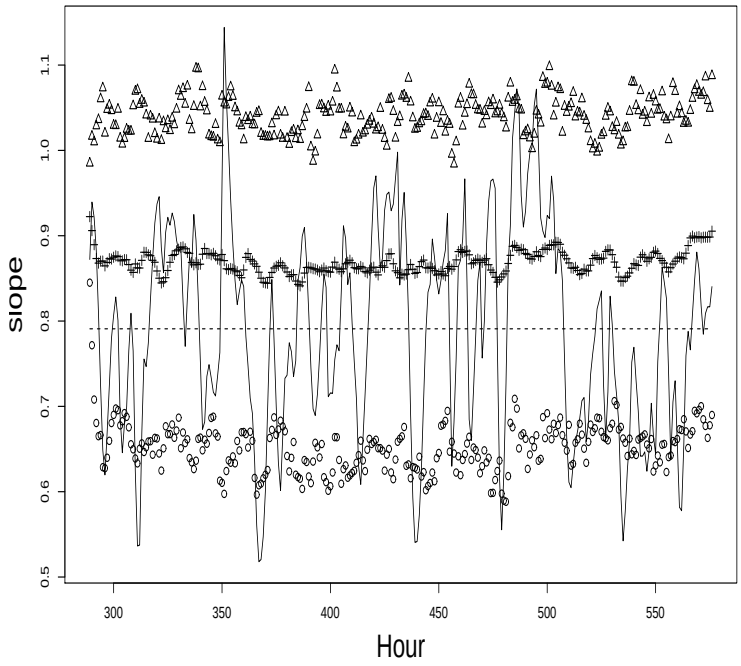

Figure 11: Rank histograms of spatial maxima [(a),(b),(c)] and space-time maxima [(d),(e),(f)]. (a)&(d). Time domain regression with block size $B = 16$. (b)&(e). Frequency domain regression with block size $B = 16$. (c)&(f). Naive approach with block size $B = 16$. All plots are for the in-sample.